Spatial clustering algorithm with obstacles constraints based on artificial bee colony

Li-ping Sun[1, 2], Yong-long Luo[1, 2*], Xin-tao Ding[2], Fu-long Chen[2]

[1] *College of National Territorial Resources and Tourism, Anhui Normal University, Anhui, 241000, China*

[2] *Engineering Technology Research Center of Network and Information Security, Anhui Normal University, Anhui, 241000, China*

Received 1 August 2014, www.cmnt.lv

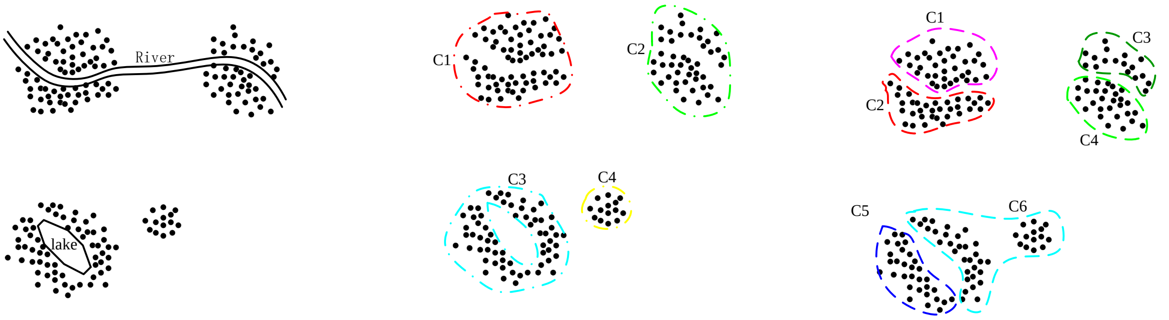
Abstract

Spatial clustering is one of practical data mining technique. In this paper, artificial bee colony (ABC) is used for clustering algorithm, which aims to optimally partition N objects into K clusters in obstacle space. The ABC algorithm used for clustering analysis with obstacles constraints, called The ABC algorithm used for clustering analysis with obstacles constraints ABC-CO, is proposed in the paper. By comparison with the two classic clustering algorithms, *k*-medoids and COE-CLARANS, demonstrates the rationality and usability of the ABC-CO algorithm.

Keywords: spatial clustering, artificial bee colony, obstructed distance; fitness calculation

1 Introduction

Spatial clustering is the organization of geographical data set into homogenous groups, the aim of which is to group spatial data points into clusters [1-3]. Most spatial clustering algorithms apply Euclidean distance between two sample points to measure the proximity of spatial points. However, physical obstacles (e.g. rivers and highways) often exist in real applications, which can hinder straight reachability among sample points. As a result, the clustering results, which utilize Euclidean distance measure are often unreasonable. Taking the simulated dataset in Figure 1a as an example, where the points represent the location of consumers. The clustering result shown in Figure 1b can be obtained, when the rivers and hill as obstacles are not considered. Obviously, the result is not realistic. If the obstacles are taken into account, the clustering result in Figure 1c can be obtained.



a) Spatial dataset with obstacles

b) Spatial clustering result ignoring obstacles

c) Spatial clustering result considering obstacles

FIGURE 1 Spatial clustering in the presence of obstacles

At present, there are a few algorithms considering obstacles constraints in the spatial clustering process [4-9]. There generally exists the shortcomings, including low robustness and easy to fall into local optimum. Many heuristic clustering algorithms have been introduced to overcome local optima problem, such as evolutionary algorithms [10, 11], swarm intelligence algorithms [12, 13] and so on [14, 15]. Artificial bee colony (ABC) algorithm, which simulates the intelligent foraging behaviour of a honey bee swarm, is a novel category of heuristic algorithms. In this paper, ABC optimization algorithm for optimization problems, which is proposed by Karaboga [16], is applied to spatial clustering analysis. Fathian and et al. proposed the application of honeybee mating optimization in clustering [17]. Zhang and et al. presented a novel artificial bee colony approach for clustering which was compared with other heuristic algorithms such as genetic algorithm, ant colony, simulated annealing and tabu search [18]. Yan and et al. designed a hybrid artificial bee colony (HABC) algorithm for data clustering [19]. Karaboga and Ozturk applied

[*] ***Corresponding author*** e-mail: ylluo@ustc.edu.cn

artificial bee colony (ABC) optimization algorithm to classification benchmark problems [20].

In this paper, ABC algorithm is extended for clustering analysis with obstacles constraints. The performance of the algorithm has been tested on a variety of simulated data sets provided from real-life situations and compared with classic clustering algorithms. The remainder of this paper can be structured as follows. Section 2 gives the related definitions and elaborates the detailed of ABC-CO algorithm. Section 3 shows the experimental results. Section 4 presents the conclusions and main findings.

2 The clustering analysis with obstacles constraints

Let $E = \{E_1, E_2, \dots, E_i, \dots, E_n\}$ represents a collection of spatial entities with autocorrelation. Let (e_{i1}, e_{i2}, ..., e_{ij}, ..., e_{im}) represents feature vector of a spatial entity E_i. Divide E into k clusters, denoted $E = \{C_1, C_2, ..., C_i, ..., C_k\}$, $C_i = \{E_1, E_2, ..., E_t\}$. $S(E_{ai}, E_{bj})$ represents the similarity of the i-th entity in the a-th cluster and j-th entity in the b-th cluster. The spatial clustering result satisfies following conditions:

1) $\bigcup_{i=1}^{k} C_i = E$;

2) For $\forall C_a, C_b \subseteq E$, $a \neq b$, the following conditions need to be satisfied simultaneously:

1) $C_a \cap C_b = \varphi$;

2) $MAX_{\forall E_{ai} \in C_a, \forall E_{bj} \in C_b} (S(E_{ai}, E_{bj})) <$
$MIN_{\forall E_{ai} \in C_a, \forall E_{bj} \in C_b} (S(E_{ai}, E_{bj}))$.

In this paper, we mainly focus on clustering analysis with obstacles constraints. The obstacles have two typical classes: convex obstacle and concave obstacle. Relevant definitions are defined as follows:

Definition 1 (Polygon obstacles). Let O be the set of m non-intersecting obstacles $\{o_1, o_2, \dots, o_m\}$. Each obstacle o_i is represented by a simple polygon, denoted as G (V, E), where V = (v_1, v_2, …, v_n) is the set of vertices of the polygon, E= {(v_k, $v_{(k+1) \bmod n}$) | k = 1,…n}, n is the number of the vertices.

Definition 2 (Reachability between two points). For $\forall$ p, q in a two-dimensional space, if segment pq does not intersect with any obstacle, p is called directly reachable from q; Otherwise, p is called indirectly reachable from q.

Definition 3 (Obstructed distance). Given point p and q, $d_o(p,q)$ represents the obstructed distance between two sample points which represent represents the entities. If p is directly reachable from q, $d_o(p,q)$ is Euclidean distance between p and q, denotes as $d(p,q)$ = $\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$. If p is indirectly reachable from q, $d_o(p,q)$ is the length of the path which is configured to bypass the obstacles while p, q respectively is taken as the start and end point.

The degree of similarity between the objects can be identified by some criterions, such as the total within-cluster variance or the total mean-square quantization error (MSE) [21], which is defined as follows:

$$Perf(O,C) = \sum_{i=1}^{n} MIN\{\|o_i - c_j\|^2, j = 1, 2, ..., k\}, \qquad (1)$$

where $\|o_i - c_j\|$ denotes the similarity between the j-th centre and the i-th object. The most used similarity metric in clustering procedure is the distance metric. In this paper, the obstructed distance described in Definition 3 is used as a distance metric. The following algorithm is the calculation of the obstructed distance between objects follows:

```
Algorithm 1: Calculation of obstructed distance
1: If ($p$ is directly reachable from $q$)
2:         $d_o(p,q) = d(p,q)$
3:         return
4: else
5:         shape = polygon_shape($G$ ($V$, $E$)) /*Judge the shape of
           the obstacles*/
6:         go 8
7: End if
8: If (shape = convex)
9:         Construct_visibility_graph($p$, $G$ ($V$, $E$), $q$) /*Constuct the
           visibility graph with obstacles by the method in Literature
           [22] */
10:        $d_o(p,q)$ = Dijkstra_Calculate_distance ($p,q$) /*Apply
           Dijkstra algorithm [23] which is the classic model to
           calculate the shortest distance of $d_o(p,q)$ based on the
           visibility graph achieved by the previous operation.*/
11: else /*Means the obstacle shape of the obstacles is concave.*/
12:        Eliminate the concave points from $V$, denoted $V'$ /*If two
           points is obstructed by concave obstacles, the concave
           points should not be linked */
13:        Construct_visibility_graph($p$, $G$ ($V'$, $E$), $q$)
14:        $d_o(p,q)$ = Dijkstra_Calculate_ distance ($p,q$)
15: End if
```

3 Application of artificial bee colony algorithm in clustering

3.1 HONEY BEE MODELLING

Artificial bee colony (ABC) algorithm was presented by Karaboga [16]. In ABC algorithm, the colony of artificial bees consists of three essential categories of bees: employed bees, onlookers and scouts. Half of the colony consists of the employed bees and the other half contains the onlookers. Corresponding to each food source, there is one employed bee. The food source which is abandoned by the bees is replaced with a new food source by the scout bees. A food source represents a possible solution to the combinatorial optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of associated solution. Pseudo-code of the ABC algorithm can be elaborated as follows:

Algorithm 2: The framework of the classic ABC algorithm.

```
Initialize the population of feasible solutions and evaluate the fitness
   of the solutions.
Generate new solutions and calculate the fitness of the solutions.
Evaluate the probabilities of preferable solutions.
Choose solutions depending on probabilities and generate new
   solutions.
Replace the abandoned solutions with new solutions.
Memorize the best solution so far.
```

In the initialization phase, the ABC algorithm generates a randomly distributed initial population of N solutions, where N denotes the size of population. Each solution $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ can be generated as follows:

$$x_{ij} = x_j^{\min} + (x_j^{\max} \quad x_j^{\min}) rand(0,1) , \tag{2}$$

where $i = 1, 2, \dots, N, j = 1, 2, \dots, D$. D is the dimension of the optimization problem. $x_j^{\min}$ and $x_j^{\max}$ are lower and upper bounds of the j-th parameter. The fitness of solutions will be calculated by Equation (3).

$$fit_i = \frac{1}{1+f_i} , \tag{3}$$

where f_i is the objective function value of the solution x_i. In the employed bee' phase, each employed bee is sent to the food source in its memory and finds a neighbouring food source $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, which is generated by Equation (4) as followed:

$$v_{i,j} = x_{i,j} + \varphi_{i,j}(x_{i,j}, x_{k,j}) , \tag{4}$$

where k is a randomly selected dimension different from i, j. $\phi_{i,j} \in [-1, 1]$ is a random number.

In the onlooker bees' phase, an onlooker selects a food source depending on the probability value of the food source, which is calculated by the following Equation:

$$p_i = \frac{fit_i}{\sum_{j=1}^{N} fit_j} . \tag{5}$$

3.2 THE ABC ALGORITHM USED FOR CLUSTERING ANALYSIS

In this paper, the ABC-CO algorithm is proposed for clustering analysis with obstacles constraints. In a clustering problem, a set of n objects need to be divided into k clusters. Let $V = \{v_1, v_2, \dots, v_n\}$ be the set of n objects. Each object has m characters. The character data matrix is denoted $D_{n\times m}$. The i-th row of the profile data matrix $D_{n\times m}$ presents an object v_i. Let $C = \{C_1, C_2, \dots , C_k\}$ be the set of k clusters, each candidate solution of the population consists of m times k cells c_{ij} ($i\in\{1, \dots, k\}$, $j\in\{1, \dots, m\}$). In order to apply the ABC algorithm to solve clustering algorithm, floating point arrays are utilized to solve cluster centres. Each food source presents a set of cluster centre, seen in Equation (6). A food source can be decoded to a cluster centre following the Equation (7).

$$X_i = \{x_1, x_2, \dots, x_m, x_{m+1}, \dots, x_{n\times m}\} , \tag{6}$$

$$C_i = \{x_{(i-1)\times m+1}, x_{(i-1)\times m+2}, \dots, x_{i\times m}\} , \quad (i = 1, 2, \dots, k) \tag{7}$$

where X_i denotes a food source in ABC-CO algorithm, k is the number of clusters and m is the number of characters of the clustering data.

The total within-cluster variance in Equation (1) is used to evaluate the quality of clusters partition for the ABC-CO algorithm. The pseudo-code of the fitness calculation of ABC-CO algorithm is shown in Algorithm 3.

Algorithm 3: Fitness calculation of a solution.

```
Perform Algorithm 1 to compute the obstructed distance between
   all objects in V and each cluster centre.
Assign objects to the nearest centres.
Calculate the total within-cluster variance $t_i$ following Equation (1).
$fit_i = t_i$
```

According to the explanation of classic ABC algorithm and the adjustments for clustering analysis with obstacles constraints, the pseudo-code of the ABC-CO algorithm is shown in Algorithm 4.

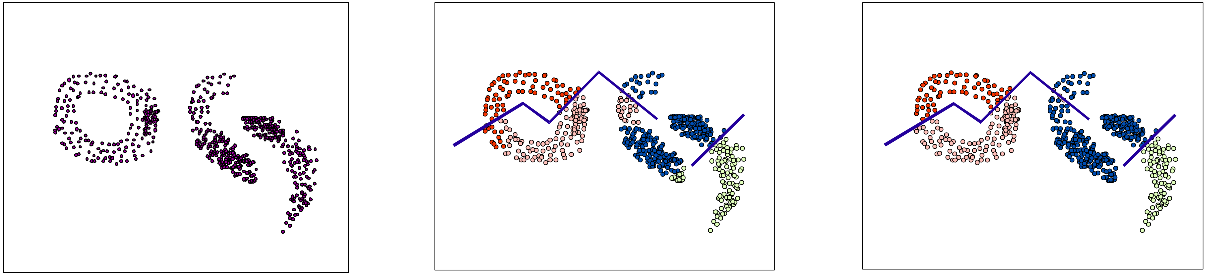
Algorithm 4: The ABC algorithm used for clustering analysis with obstacles constraints (ABC-CO).

```
Set the population size N, the control parameter limit and the
   maximum cycle number MCN.
Randomly generate the initial population of solutions, which can
   be decoded to the cluster centres following Equation (7).
For (each food source $C_i$)
    Perform Algorithm 3 to calculate the fitness of $C_i$
End for
Set $trial_i = 0$, where i =1, 2 , …, N and iteration = 1
While (iteration ≤ MCN)
    For (each food source $C_i$)
        Generate a new solution $v_i$ following the Equation(7);
        Perform Algorithm 3 to calculate the fitness of $v_i$
        If (the fitness of $v_i$ is better than or equal to the fitness of $C_i$)
            then $trial_i = 0$, otherwise $trial_i = trial_i + 1$
    End for
    Evaluate the probability $p_i$ for each solution following the
        Equation (5);
    Set I=1 and the current onlooker bee number ln = 0
    While (ln < N)
        If (rand() < $p_i$)
            Generate a new solution $v_j$ following the Equation (7)
            Perform Algorithm 3 to calculate the fitness of $v_j$
            Apply greedy selection process on the original solution and the
                new solution
            ln = ln + 1
        End if
        Set i = i + 1and if i > N then i =1
    End while
    For (each food source $C_i$)
        If ($trial_i$ ≥ limit)
            Replace the abandoned solution with a new solution which is
                randomly generated by Equation (7)
        End if
    End for
    Memorize the best solution so far
    Set iteration = iteration +1
End while
```

4 Results and discussion

This paper presents two sets of experiments to prove the effectiveness of the ABC-CO algorithm: The first experiment uses a set of analogue data, which are generated by the simulation of ArcGIS 9.3. Experimental results are compared with *k*-medoids clustering algorithm [2]. The second experiment carries on spatial dataset, and compares the results with the COE-CLARANS algorithm [8]. All algorithms are implemented in C# language and executed on a Pentium 4.3*HZ*, 2*GB* RAM computers. The population size *N*=40, the maximum cycle number *MCN*=2000 and the parameter *limit*=100.

The classic *k*-medoids clustering algorithm has been widely used for its simplicity and feasibility. ABC-CO algorithm uses obstacle distance defined in this paper for clustering analysis, and *k*-medoids algorithm uses Euclidean distance as similarity measure of samples. Simulated dataset of the first experiment are shown in Figure 2a. When cluster number *k*=6, the clustering results of *k*-medoids clustering algorithm and ABC-CO algorithm are shown in Figures 2b and 2c, respectively. Experimental results show that the clustering results of the ABC-CO algorithm considering obstacles are more efficient than *k*-medoids algorithm. This paper presents two sets of experiments to prove the effectiveness of the ABC-CO algorithm: The first experiment uses a set of analog data, which are generated by the simulation of ArcGIS 9.3. Experimental results are compared with *k*-medoids clustering algorithm [2]. The second experiment carries on spatial dataset, and compares the results with the COE-CLARANS algorithm [8]. All algorithms are implemented in C# language and executed on a Pentium 4.3*HZ*, 2*GB* RAM computers. the population size *N*=40, the maximum cycle number *MCN*=2000 and the parameter *limit*=100.



a) Simulated dataset b) Clustering results of *k*-medoids algorithm c) Clustering results of ABC-CO algorithm

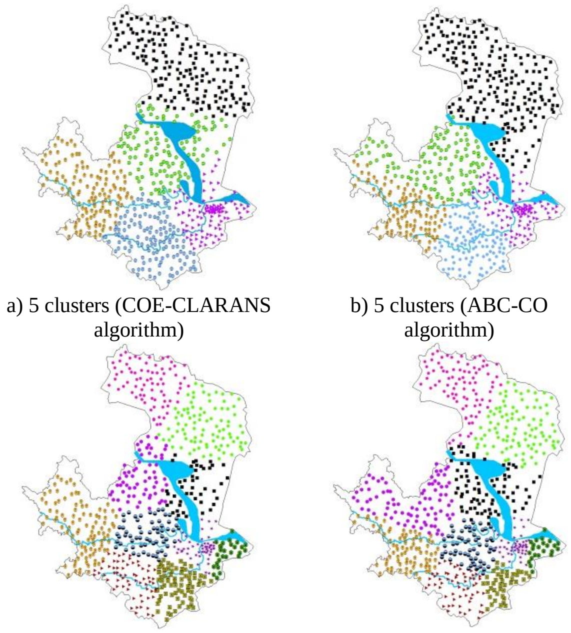
FIGURE 2 Clustering results for analog data

4.1 SIMULATION EXPERIMENT

The classic *k*-medoids clustering algorithm has been widely used for its simplicity and feasibility. ABC-CO algorithm uses obstacle distance defined in this paper for clustering analysis, and *k*-medoids algorithm uses Euclidean distance as similarity measure of samples. Simulated dataset of the first experiment are shown in Figure 2a. When cluster number *k*=6, the clustering results of *k*-medoids clustering algorithm and ABC-CO algorithm are shown in Figures 2b and 2c, respectively. Experimental results show that the clustering results of the ABC-CO algorithm considering obstacles are more efficient than *k*-medoids algorithm.

4.2 CLUSTERING ALGORITHM APPLICATION AND CONTRASTIVE ANALYSIS

This paper takes resident communities as cluster points, where the points are represented as (x, y). The hills, rivers and lakes in the territory are spatial obstacles which are described in Definition 1, denoted as G (V, E). Based on above real world datasets, the COE-CLARANS algorithm and the ABC-CO algorithm are compared by simulation experiment. The results are shown in Figure 3:



a) 5 clusters (COE-CLARANS algorithm) b) 5 clusters (ABC-CO algorithm)

c) 10 clusters (COE-CLARANS algorithm) d) 10 clusters (ABC-CO algorithm)

FIGURE 3 Comparison of clustering analysis using COE-CLARANS algorithm and ABC-CO algorithm

In view of the range of radiation area of different type public facility, the paper has carried on the cluster simulation deferred to 5 subclasses and 10 subclasses separately. For Yangtze River is the main obstacle of Wuhu territory, the result of its peripheral region clustering analysis can demonstrate the validity of the algorithm. As can be seen from the clustering results, the subclasses that the COE-CLARANS algorithm obtains are more centralized. And the community intensive places are divided by obstacles in several subclasses. The actual distance between a cluster sample point and centre, which do not belong to the same side of an obstacle is much farther than the straight line distance between them. The ABC-CO algorithm has solved this problem well. The results of the algorithm demonstrate subclasses of clustering divided by physical obstacles is quite few, which divide at the community sparse place.

5 Conclusions

Spatial clustering analysis is an important method of in the data mining community. Traditional clustering methods use a variety of straight-line distance metrics to measure the degree of similarity between spatial entities. In this paper, an artificial bee colony algorithm is extended to solve clustering problems with obstacles constraints, which is inspired by the bees' forage behaviour. This algorithm was implemented and tested on several real spatial datasets. By comparison with the classic clustering algorithms, it verifies the rationality and usability of the ABC-CO algorithm.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61370050) and Natural Science Foundation of Anhui Province (No. 1308085QF118).

References

[1] Oliveira D, Jr J 2011 *Advanced Engineering Informatics* **25** 380-9

[2] Wang Z, Soh Y C, Song Q, Sim K 2009 *Pattern Recognition* **42** 2029-44

[3] Pennerstorfer D, Weiss C 2013 *Regional Science and Urban Economics* **43** 661-75

[4] Tung A, Hou J, Han J 2000 *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining* 165-8

[5] Ng R, Han J 1994 *Proceedings of the 20th International Conference on Very Large Data Bases* 144-55

[6] Zaïane O R, Lee C H 2002 *Proceedings of the IEEE International Conference on Data Mining* 737-40

[7] Ester M, Kriegel H, Sander J 1996 *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* 226-31

[8] Wang X, Rostoker C, Hamilton H 2012 *Journal of Intelligent Information Systems* **38** 269-97

[9] Estivill-Castro V, Lee I 2004 *Journal of Photogrammetry and Remote Sensing* **59** 21-34

[10] Jiang B, Wang N, Wang L P 2013 *Communications in Nonlinear Science and Numerical Simulation* **18**(3)134-45

[11] Cura T 2012 *Expert Systems with Applications* **39** 1582-8

[12] Bereta M, Burczyński T 2009 *Information Sciences* **179** 1407-25

[13] Kuo R J, Lin L M 2010 *Decision Support Systems* **49** 45-62

[14] Gou S P, Zhuang X, Li Y Y, Xu C, Jiao L C 2013 *Neurocomputing* **4** 275-89

[15] Liu R C, Zhang X G, Yang N, Lei Q F, Jiao L C 2012 *Applied Soft Computing* **12** 302-12

[16] Karaboga D 2005 Technical Report-TR06 *Erciyes University Engineering Faculty*

[17] Fathin M, Amiri B, Maroosi A 2007 *Applied Mathematics and Computation* **190** 1502-13

[18] Zhang C S, Ouyang D T, Ning J X 2010 *Expert Systems with Applications* **37** 4761-67

[19] Yan X H, Zhu Y L, Zou W P 2012 *Neurocomputing* **97** 241-50

[20] Karaboga D, Ozturk C 2011 *Applied Soft Computing* **11** 652-7

[21] Gűngőr Z, Űnler A 2007 *Applied Mathematics and Computation* **184(2)** 199-209

[22] Lozano-Perez T, Wesley M 1979 *Communications of the ACM* **22** 436-50

[23] Marianov V, Revelle C 1996 *European Journal of Operational Research* **93** 110-20

Authors

Li-ping Sun, born in June, 1980, Anhui, China

Current position: associate professor in Anhui Normal University.
University studies: Computer Software and Theory in Chongqing University.
Scientific interest: computer science, computer modelling and spatial data mining.

Yong-long Luo, born in April, 1972, Anhui, China

Current position: professor in Anhui Normal University.
University studies: Computer Science and Technology in University of Science and Technology of China.
Scientific interest: computer modelling, network security, applied mathematics and spatial data mining.

Xin-tao Ding, born in December, 1979, Anhui, China

Current position: lector in Anhui Normal University.
University studies: Computational Mathematics in East China Normal University.
Scientific interest: computer science, computer modelling and spatial data mining.

Fu-long Chen, born in May, 1978, Anhui, China

Current position: associate professor in Anhui Normal University.
University studies: Computer Science and Technology in Northwestern Polytechnical University.
Scientific interest: computer science, computer modelling and spatial data mining.